# CH55X 汇编指令说明

适用型号包含：**CH551、CH552、CH553、CH554、CH555、CH557、CH558、CH559**

CH55X汇编指令周期概述：

a. 非跳转指令的指令周期数与指令字节数相同（除了CH551/2/3/4的DIV指令）；

b. 大多数指令都是单字节、1T单周期指令；

c. 跳转指令含MOVC/RET/CALL通常比指令字节数多若干个周期；

d. RET/RETI指令多3个周期起，如返回地址为奇数再加1周期；

e. MOVC指令多4个周期起，如本指令地址为奇数再加1周期，如下条指令地址为奇数再加1周期；

f. JMP@A指令多2个周期起，如本指令地址为奇数再加1周期，如目标地址为奇数再加1周期；

g. JB/JNB/JBC/CJNE_A,dir/DJNZ指令多2个周期起，如本指令地址为奇数再加1周期，如目标地址为奇数再加1周期；

h. 其余跳转指令多2个周期起，如目标地址为奇数再加1周期；

i. 条件跳转指令如果未发生跳转则周期数与指令字节数相同。

注：以上周期是指当前 CH55X 的系统主频的倒数。

| 类别 | 指令格式 | 功能描述 | 指令字节 | 指令周期 |
|---|---|---|---|---|
| 传送类指令（29条） | MOV A,Rn | (Rn)→(A) Rn 中的内容送到累加器 A 中，Rn=R0-R7 | 1 | 1 |
| | MOV A,direct | (direct)→(A) 直接单元地址中的内容送到累加器 A | 2 | 2 |
| | MOV A,@Ri | (Ri)→(A) Ri 内容指向的地址单元中的内容送到累加器 A，Ri=R0 或 R1 | 1 | 1 |
| | MOV A,#data | #data→(A) 立即数送到累加器 A 中 | 2 | 2 |
| | MOV Rn,A | (A)→(Rn) 累加器 A 中的内容送到寄存器 Rn 中 | 1 | 1 |
| | MOV Rn,direct | (direct)→(Rn) 直接寻址单元中的内容送寄存器 | 2 | 2 |
| | MOV Rn,#data | #data→(Rn) 立即数直接送到寄存器 Rn 中 | 2 | 2 |
| | MOV direct,A | (A)→(direct) 累加器 A 中的内容送直接寻址单元 | 2 | 2 |
| | MOV direct,Rn | (Rn)→(direct) 寄存器中的内容送直接寻址单元 | 2 | 2 |
| | MOV direct2,direct1 | (direct1)→(direct2) 直接寻址单元中的内容 1 送直接寻址单元 2 | 3 | 3 |
| | MOV direct,@Ri | (Rn)→(direct) 寄存器中的内容送直接寻址单元 | 2 | 2 |
| | MOV direct,#data | #data→(direct) 立即数送直接寻址单元 | 3 | 3 |
| | MOV @Ri,A | (A)→((Ri)) 累加器 A 中的内容送到以 Ri 中的内容为地址的 RAM 单元 | 1 | 1 |
| | MOV @Ri,direct | (direct)→((Ri)) 直接寻址单元内容送到以 Ri 中的内容为地址的 RAM 单元 | 2 | 2 |
| | MOV @Ri,#data | #data→((Ri)) 立即数送到以 Ri 中的内容为地址的 RAM 单元 | 2 | 2 |
| | MOV DPTR,#data16 | #dataH→(DPH)，#dataL→(DPL) 16 位常数的高 8 位送到 DPH，低 8 位送到 DPL | 3 | 3 |
| | MOVX A,@DPTR | ((DPTR))→(A) 数据指针指向外部 RAM(16 位地址)的内容送到累加器 A | 1 | 1 |
| | MOVX A,@Ri | ((Ri))→(A) 寄存器 Ri 指向外部 RAM(8 位地址)的内容送到累加器 A 中，高 8 位地址由 P2 提供 | 1 | 1 |
| | MOVX @DPTR,A | (A)→((DPTR)) 累加器中的内容送到数据指针指向外 | 1 | 1 |

| | | | | |
|---|---|---|---|---|
| | | 部 RAM(16 位地址) | | |
| | MOVX @Ri,A | (A)→((Ri)) 累加器中的内容送到寄存器 Ri 指向外部 RAM(8 位地址），高 8 位地址由 P2 提供 | 1 | 1 |
| | MOVC A,@A+DPTR | ((A)+(DPTR))→(A) 表格地址单元中的内容送到累加器 A 中 | 1 | 5/6/7 |
| | MOVC A,@A+PC | (PC)+1→(PC)，((A)+(PC))→(A) 表格地址单元中的内容送到累加器 A 中 | 1 | 5/6/7 |
| | XCH A,Rn | (A)←→(Rn) 累加器与工作寄存器 Rn 中的内容互换 | 1 | 1 |
| | XCH A,direct | (direct)→(A) 累加器 A 的内容与直接寻址单元的内容交换 | 2 | 2 |
| | XCH A,@Ri | (A)←→((Ri)) 累加器与工作寄存器 Ri 所指的存储单元中的内容互换 | 1 | 1 |
| | XCHD A,@Ri | (A[3:0])←→((Ri)[3:0]) 累加器与工作寄存器 Ri 所指的存储单元中的内容低半字节互换 | 1 | 1 |
| | SWAP A | (A[3:0])←→(A[7:4]) 累加器中的内容高低半字节互换 | 1 | 1 |
| | PUSH direct | (SP)+1→(SP)，(direct)→(SP) 堆栈指针首先加 1，直接寻址单元中的数据送到堆栈指针 SP 所指的单元中 | 2 | 2 |
| | POP direct | (SP)→(direct)，(SP)-1→(SP) 堆栈指针 SP 所指的单元数据送到直接寻址单元中，堆栈指针 SP 再进行减 1 操作 | 2 | 2 |
| 算术运算类指令(24 条) | ADD A,Rn | (A)+(Rn)→(A) 累加器 A 中的内容与工作寄存器 Rn 中的内容相加，结果存在 A 中 | 1 | 1 |
| | ADD A,direct | (A)+(direct)→(A) 累加器 A 中的内容与直接地址单元中的内容相加，结果存在 A 中。 | 2 | 2 |
| | ADD A,@Ri | (A)+((Ri))→(A) 累加器 A 中的内容与工作寄存器 Ri 所指向地址单元中的内容相加，结果存在 A 中 | 1 | 1 |
| | ADD A,#data | (A)+#data→(A) 累加器 A 中的内容与立即数#data 相加，结果存在 A 中 | 2 | 2 |
| | ADDC A,Rn | (A)+Rn+(C)→(A) 累加器 A 中的内容与工作寄存器 Rn 中的内容、连同进位位相加，结果存在 A 中 | 1 | 1 |
| | ADDC A,direct | (A)+(direct)+(C)→(A) 累加器 A 中的内容与直接地址单元的内容连同进位位相加，结果存在 A 中 | 2 | 2 |
| | ADDC A,@Ri | (A)+((Ri))+(C)→(A) 累加器 A 中的内容与工作寄存器 Ri 指向地址单元中的内容、连同进位位相加，结果存在 A 中 | 1 | 1 |
| | ADDC A,#data | (A)+#data+(C)→(A) 累加器 A 中的内容与立即数连同进位位相加，结果存在 A 中 | 2 | 2 |
| | INC A | (A)+1→(A) 累加器 A 中的内容加 1，结果存在 A 中 | 1 | 1 |
| | INC Rn | (Rn)+1→(Rn) 寄存器 Rn 的内容加 1，结果送回原地址单元中 | 1 | 1 |
| | INC direct | (direct)+1→(direct) 直接地址单元中的内容加 1，结果送回原地址单元中 | 2 | 2 |
| | INC @Ri | ((Ri))+1→((Ri)) 寄存器的内容指向的地址单元中的内容加 1，结果送回原地址单元中 | 1 | 1 |

| | | | | |
|---|---|---|---|---|
| | INC DPTR | (DPTR)+1→(DPTR)数据指针的内容加 1，结果送回数据指针中 | 1 | 1 |
| | SUBB A,Rn | (A)-(Rn)-(C)→(A) 累加器 A 中的内容与工作寄存器中的内容、连同借位位相减，结果存在 A 中 | 1 | 1 |
| | SUBB A,direct | (A)-(direct)-(C)→(A) 累加器 A 中的内容与直接地址单元中的内容、连同借位位相减，结果存在 A 中 | 2 | 2 |
| | SUBB A,@Ri | (A)-((Ri))-(C)→(A) 累加器 A 中的内容与工作寄存器 Ri 指向的地址单元中的内容、连同借位位相减，结果存在 A 中 | 1 | 1 |
| | SUBB A,#data | (A)-#data-(C)→(A) 累加器 A 中的内容与立即数、连同借位位相减，结果存在 A 中 | 2 | 2 |
| | DEC A | (A)-1→(A)累加器 A 中的内容减 1，结果送回累加器 A 中 | 1 | 1 |
| | DEC Rn | (Rn)-1→(Rn)寄存器 Rn 中的内容减 1，结果送回寄存器 Rn 中 | 1 | 1 |
| | DEC direct | (direct)-1→(direct)直接地址单元中的内容减 1，结果送回直接地址单元中 | 2 | 2 |
| | DEC @Ri | ((Ri))-1→((Ri))寄存器 Ri 指向的地址单元中的内容减 1，结果送回原地址单元中 | 1 | 1 |
| | MUL AB | (A)×(B)→(A)和(B) 累加器 A 中的内容与寄存器 B 中的内容相乘，结果存在 A、B 中 | 1 | 1 |
| | DIV AB (CH555/557/558/559) | (A)÷(B)→(A)和(B) 累加器 A 中的内容除以寄存器 B 中的内容，所得到的商存在累加器 A，而余数存在寄存器 B 中 | 1 | 1 |
| | DIV AB (CH551/552/553/554) | (A)÷(B)→(A)和(B) 累加器 A 中的内容除以寄存器 B 中的内容，所得到的商存在累加器 A，而余数存在寄存器 B 中 | 1 | 4 |
| | DA A | 累加器进行 10 进制转换 | 1 | 1 |
| 逻辑操作指令（24 条） | ANL A,Rn | 累加器 A 的内容和寄存器 Rn 中的内容执行与逻辑操作，结果存在累加器 A 中 | 1 | 1 |
| | ANL A,direct | 累加器 A 中的内容和直接地址单元中的内容执行与逻辑操作，结果存在寄存器 A 中 | 2 | 2 |
| | ANL A,@Ri | 累加器 A 的内容和工作寄存器 Ri 指向的地址单元中的内容执行与逻辑操作，结果存在累加器 A 中 | 1 | 1 |
| | ANL A,#data | 累加器 A 的内容和立即数执行与逻辑操作。结果存在累加器 A 中 | 2 | 2 |
| | ANL direct,A | 直接地址单元中的内容和累加器 A 的内容执行与逻辑操作，结果存在直接地址单元中 | 2 | 2 |
| | ANL direct,#data | 直接地址单元中的内容和立即数执行与逻辑操作，结果存在直接地址单元中 | 3 | 3 |
| | ORL A,Rn | 累加器 A 的内容和寄存器 Rn 中的内容执行逻辑或操作，结果存在累加器 A 中 | 1 | 1 |
| | ORL A,direct | 累加器 A 中的内容和直接地址单元中的内容执行逻辑或操作，结果存在寄存器 A 中 | 2 | 2 |
| | ORL A,@Ri | 累加器 A 的内容和工作寄存器 Ri 指向的地址单元中的 | 1 | 1 |

| | | | | |
|---|---|---|---|---|
| | | 内容执行逻辑或操作，结果存在累加器 A 中 | | |
| | ORL A,#data | 累加器 A 的内容和立即数执行逻辑或操作，结果存在累加器 A 中 | 2 | 2 |
| | ORL direct,A | 直接地址单元中的内容和累加器 A 的内容执行逻辑或操作，结果存在直接地址单元中 | 2 | 2 |
| | ORL direct,#data | 直接地址单元中的内容和立即数执行逻辑或操作，结果存在直接地址单元中 | 3 | 3 |
| | XRL A,Rn | 累加器 A 的内容和寄存器 Rn 中的内容执行逻辑异或操作，结果存在累加器 A 中 | 1 | 1 |
| | XRL A,direct | 累加器 A 中的内容和直接地址单元中的内容执行逻辑异或操作，结果存在寄存器 A 中 | 2 | 2 |
| | XRL A,@Ri | 累加器 A 的内容和工作寄存器 Ri 指向的地址单元中的内容执行逻辑异或操作，结果存在累加器 A 中 | 1 | 1 |
| | XRL A,#data | 累加器 A 的内容和立即数执行逻辑异或操作，结果存在累加器 A 中 | 2 | 2 |
| | XRL direct,A | 直接地址单元中的内容和累加器 A 的内容执行逻辑异或操作，结果存在直接地址单元中 | 2 | 2 |
| | XRL direct,#data | 直接地址单元中的内容和立即数执行逻辑异或操作，结果存在直接地址单元中 | 3 | 3 |
| | CLR A | 0→(A)，累加器中的内容清 0 | 1 | 1 |
| | CPL A | 累加器中的内容按位取反 | 1 | 1 |
| | RL A | 累加器 A 中的内容左移一位 | 1 | 1 |
| | RR A | 累加器 A 中的内容右移一位 | 1 | 1 |
| | RLC A | 累加器 A 中的内容连同进位位 CY 左移一位 | 1 | 1 |
| | RRC A | 累加器 A 中的内容连同进位位 CY 右移一位 | 1 | 1 |
| 控制转移指令（16 条） | LJMP addr16 | addr16→(PC)，给程序计数器赋予新值(16 位地址) | 3 | 5/6 |
| | AJMP addr11 | (PC)+2→(PC)，addr11→(PC[10:0])程序计数器赋予新值(11 位地址)，(PC[15:11])不改变。 | 2 | 4/5 |
| | SJMP rel | (PC)+2+rel→(PC)当前程序计数器先加上 2 再加上偏移量给程序计数器赋予新值。 | 2 | 4/5 |
| | JMP @A+DPTR | (A)+(DPTR)→(PC)，累加器所指向地址单元的值加上数据指针的值给程序计数器赋予新值。 | 1 | 3/4/5 |
| | JZ rel | A=0,(PC)+2+rel→(PC),累加器中的内容为 0，则转移到偏移量所指向的地址，否则程序往下执行 | 2 | 2 或 4/5 |
| | JNZ rel | A≠0,(PC)+2+rel→(PC),累加器中的内容不为 0,则转移到偏移量所指向的地址，否则程序往下执行 | 2 | 2 或 4/5 |
| | CJNE A,#data,rel | A≠#data,(PC)+3+rel→(PC),累加器中的内容不等于立即数，则转移到偏移量所指向的地址，否则程序往下执行 | 3 | 3 或 5/6 |
| | CJNE Rn,#data,rel | A≠#data,(PC)+3+rel→(PC),工作寄存器 Rn 中的内容不等于立即数，则转移到偏移量所指向的地址，否则程序往下执行 | 3 | 3 或 5/6 |
| | CJNE @Ri,#data,rel | A≠#data,(PC)+3+rel→(PC),工作寄存器 Ri 指向地址单元中的内容不等于立即数，则转移到偏移量所指向的地址，否则程序往下执行 | 3 | 3 或 5/6 |

| | | | | |
|---|---|---|---|---|
| | CJNE A,direct,rel | A≠(direct)，(PC)+3+rel→(PC)，累加器中的内容不等于直接地址单元的内容，则转移到偏移量所指向的地址，否则程序往下执行 | 3 | 3 或 5/6/7 |
| | DJNZ Rn,rel | (Rn)-1→(Rn)，(Rn)≠0，(PC)+2+rel→(PC) 工作寄存器 Rn 减 1 不等于 0，则转移到偏移量所指向的地址，否则程序往下执行 | 2 | 2 或 4/5/6 |
| | DJNZ direct,rel | (direct)-1→(direct)，(direct)≠0，(PC)+2+rel→(PC) 直接地址单元中的内容减 1 不等于 0，则转移到偏移量所指向的地址，否则程序往下执行 | 3 | 3 或 5/6/7 |
| | LCALL addr16 | 长调用指令，可在 64KB 空间调用子程序。此时(PC)+3→(PC)，(SP)+1→(SP)，(PC[7:0])→(SP)，(SP)+1→(SP)，(PC[15:8])→(SP)，addr16→(PC)，即分别从堆栈中弹出调用子程序时压入的返回地址。 | 3 | 5/6 |
| | ACALL addr11 | 绝对调用指令，可在 2KB 空间调用子程序，此时(PC)+2→(PC)，(SP)+1→(SP)，(PC[7:0])→(SP)，(SP)+1→(SP)，(PC[15:8])→(SP)，addr11→(PC[10:0]) | 2 | 4/5 |
| | RET | 子程序返回指令 | 1 | 4/5 |
| | RETI | 中断返回指令，RETI 指令不能用 RET 代替 | 1 | 4/5 |
| 位操作指令（18 条） | MOV C,bit | bit→CY，某位数据送 CY | 2 | 2 |
| | MOV bit,C | CY→bit，CY 数据送某位 | 2 | 2 |
| | CLR C | 0→CY，清 CY | 1 | 1 |
| | CLR bit | 0→bit，清某一位 | 2 | 2 |
| | SETB C | 1→CY，置位 CY | 1 | 1 |
| | SETB bit | 1→bit，置位某一位 | 2 | 2 |
| | CPL C | CY 取反 | 1 | 1 |
| | CPL bit | bit 取反 | 2 | 2 |
| | ANL C,bit | CY 和 bit 相与，结果存放到 CY | 2 | 2 |
| | ANL C,/bit | CY 和 bit 的反码相与，结果存放到 CY | 2 | 2 |
| | ORL C,bit | CY 和 bit 相或，结果存放到 CY | 2 | 2 |
| | ORL C,/bit | CY 和 bit 的反码相或，结果存放到 CY | 2 | 2 |
| | JC rel | (CY)=1 转移，(PC)+2+rel→PC，否则程序往下执行，(PC)+2→PC | 2 | 2 或 4/5 |
| | JNC rel | (CY)=0 转移，(PC)+2+rel→PC，否则程序往下执行，(PC)+2→PC | 2 | 2 或 4/5 |
| | JB bit,rel | 位状态为 1 转移，(PC)+3+rel→PC，否则程序往下执行，(PC)+3→PC | 3 | 3 或 5/6/7 |
| | JNB bit,rel | 位状态为 0 转移，(PC)+3+rel→PC，否则程序往下执行，(PC)+3→PC | 3 | 3 或 5/6/7 |
| | JBC bit,rel | 位状态为 1 转移，并使该位清 0，(PC)+3+rel→PC，否则程序往下执行，(PC)+3→PC | 3 | 3 或 5/6/7 |
| | NOP | 空指令，这条指令除了使 PC 加 1，消耗一个周期外，没有执行任何操作，可用于短时间的延时 | 1 | 1 |
| 新增指令 | DB 0xA5 | XRAM 快速复制指令 | 1 | 1 |

注：标记为灰色，代表指令字节数不等于执行该指令需要的周期

新增指令使用示例：

```
;New Instruction:   MOVX @DPTR1,A
;Instruction Code:  0xA5
;Instruction Cycle: 1
;Instruction Operation:
   ;step-1. write ACC into xdata SRAM @DPTR1 embedded chip
   ;step-2. increase DPTR1
ASM example:
       INC  XBUS_AUX
       MOV  DPTR,#TARGET_ADDR ;DPTR1
       DEC  XBUS_AUX
       MOV  DPTR,#SOURCE_ADDR ;DPTR0
       MOV  R7,#xxH
 LOOP: MOVX A,@DPTR           ;DPTR0
       INC  DPTR              ;DPTR0, if need
       DB   0A5H              ;MOVX @DPTR1,A & INC DPTR1
       DJNZ R7,LOOP
```